An Evaluation Tool for Physical Attacks

Hélène Le Bouder[1], Gaël Thomas[3], Ronan Lashermes[4], Yanis Linge[5], Bruno Robisson[2], and Assia Tria[2]

[1] IMT-Atlantique, SRCD, Cesson-Sévigné, France.
[2] EMSE, Centre de Microélectronique de Provence, Gardanne France.
[3] DGA Maîtrise de l'Information, Bruz, France.
[4] INRIA,High Security Laboratory, Rennes, France.
[5] STMicroelectronics, Rousset, France.

Abstract. The security issues of devices, used in the Internet of Things (IoT) for example, can be considered in two contexts. On the one hand, these algorithms can be proven secure mathematically. On the other hand, physical attacks can weaken the implementation. In this work, we want to compare these attacks between them. A tool to evaluate and compare different physical attacks, by separating the theoretical attack path and the experimental parts of the attacks, is presented.

1 Introduction

When talking about the security of a device, numerous tools allow the developers to prove the security of algorithms and the software design. Unfortunately, physical attacks introduce another dimension: the interaction of the implemented algorithm with the physical environment. Physical attacks are a real threat, even for algorithms proved secure mathematically. They are divided in two families: the Side Channel Analysis (SCA) and the Fault Injection Attacks (FIA).

SCA are based on observations of the circuit behaviour during the computation. The first attacks were the Simple, the Differential and the Correlation, Power Analysis (SPA, DPA and CPA) [1,2,3]. SCA exploit the fact that some physical values of a circuit depend on intermediary values of the computation. This is the so-called leakage of information of the circuit. Leakage examples are timing [4], power consumption [5] and electromagnetic emissions (EM) [6].

FIA consist in disturbing the circuit behaviour in order to alter the correct progress of the algorithm [7,8]. Faults are injected into the device using various means such as laser [9], clock glitches [10], spikes on the power supply or electromagnetic perturbations [11].

Motivation and Contribution: The question that naturally arises is: how to evaluate and compare all physical attacks? Several works have been proposed to describe them with a common framework [12,13,14]. However, these works only cover SCA. In [15,16], the authors propose to write various SCA as DPA, a lot of work has been done to compare distinguishers as in [17]. Likewise, there are frameworks for FIA [18,19,20]. In [21], Standaert *et al.* underline the interface between theory and practice for SCA, our work enlarges this vision for both

families. The improvement of our paper is to unify the evaluation for the two families (SCA and FIA). A new tool evaluates the theoretical attacks separately form the real practical attacks.

2 Description of the attacks in three steps

Physical attacks are decomposed in the following 3-step process. The target noted K is the goal of the attack, its domain of definition is noted $\mathbb{K}$.

Step 1: Campaign. An experiment $\mathcal{E}$ is a pair (O_S, O_R) (S for Stimuli and R for Reaction) of **observables**, taken during the execution of an algorithm. A set of n experiments is called a campaign. The observables could be data as plaintext, ciphertext, faulty ciphertext; or physical measurements as EM traces, power traces, signal provided by a micro-probe, computation time *etc.* The **attack path** is an exploitable relation $\mathcal{R}$ between the observables and the target $\mathcal{K}$.

$$O_R = \mathcal{R}(O_S, K) \quad . \tag{1}$$

This relation is composed of **physical functions** f and algorithm functions. The physical functions f cannot always be described with a mathematical expression since they are often non deterministic. There is often only one physical function.

Step 2: Predictions. In the attack path $\mathcal{R}$ there are two unknowns: the target K and the physical functions f. The attacker make **guesses** k on the target K. The good guess is noted $\hat{k}$. A divide and conquer approach is generally chosen. The domain of definition of the target, $\mathbb{K}$, should be short enough so that all guesses can be tested. As already pointed out, physical functions do not always have a mathematical expression. But they can be approximated by mathematical functions called **models** or by a phase of characterization called **template** as in [22]. In FIA, models are called error functions and leakage functions in SCA. Several models m can be tested for one physical function f. Commonly, one or a small set of models is used. Finally, **predictions** are built with the attack path for each guess k on the secret, where the physical functions are replaced by models.

$$P_{m,k} = \mathcal{R}_m(O_S, k) \tag{2}$$

Step 3: Confrontation. For each hypothesis k and a model m, $P_{m,k}$ is confronted to the observables O_R with a distinguisher. A **distinguisher** is a statistical tool which is able to find the correct guess on the target. The distinguisher highlights links between physical function f and mathematical model m, they are based on different statistical criteria. $P_{m,k}$ and O_R can be considered as random variables. The distinguisher returns the guess k_d, if $k_d = \hat{k}$ the attack succeeds.

3 The evaluate and compare tool

Generally different distinguishers are compared on a same device; or different campaigns with a same distinguisher; or different models with a same

distinguisher. This paper presents a different approach. First, in a theoretical study, the models are evaluated independently from the physical functions; *i.e* $\mathcal{R}_m$ in equation (2) is studied. Then the adequacy of the models with respect to the physical functions is evaluated.

Evaluation of the theoretical attack. The set of predictions has a cardinal p. One has to remark that it is possible that $p \neq card(\mathbb{K})$. Indeed two guesses can have the same prediction during an attack. Let Θ_m be an oracle associated with a model m as illustrated in Fig. 1(a). The oracle Θ_m returns $P_{m,\hat{k}}$ the prediction which corresponds to the good guess $\hat{k}$ under model m for an chosen observable O_s. The required number of queries (on average) to Θ_m in order to retrieve the target K is noted q. It is a measure of how efficient is the theoretical attack path R_m. An oracle can combine sets of models.

Evaluation of the attack in practice. This section deals with the link between observables O_R and the good prediction $P_{m,\hat{k}}$. More precisely the two codomains, for the attack path $\mathcal{R}$ and for $\mathcal{R}_m$ are compared. Additionally there is not necessarily a bijection between the codomains (the prediction $P_{m,\hat{k}}$ and the observables O_R), as it has already been shown in [12,21,16].

A contingency table is filled with the results of n experiments. All the possible values of $P_{m,\hat{k}}$ are noted P_i, $i \in [\![1, p]\!]$ and the possible values of O_R are noted O_j, $j \in [\![1, o]\!]$. For each experiment $\mathcal{E} = (O_S, O_R)$, the reaction O_R is stored and $P_{m,\hat{k}}$ is computed. Then in the contingency table, shown in Table 1, the value at the corresponding row i (prediction is equal to P_i) and the column j (reaction is equal to O_j) is incremented. At the end, the value $a_{i,j}$ is the number of times the attacker computed the prediction P_i in conjunction with the measurement of the reaction O_j, *i.e.* the number of experiments with $P_{m,\hat{k}} = P_i$ and $O_R = O_j$.

Up to normalization by a factor n, this contingency table can be understood as the joint distribution of $P_{m,k}$ and O_R. Given an experiment (O_S, O_R), the correct prediction $P_{m,\hat{k}}(O_S)$ is given by $\Theta_m(O_S)$ and does not depend on O_R. Denote by $\hat{\imath}$, the row index corresponding to $P_{m,\hat{k}}$, so that $P_{m,\hat{k}} = P_{\hat{\imath}}$; and by $\hat{\jmath}$, the column index corresponding to the observed $O_R = O_{\hat{\jmath}}$. The probability of guessing the correct prediction $P_{m,\hat{k}}$ with O_R is:

$$\mathcal{P}\left(P_{m,\hat{k}}|O_R\right) = \frac{a_{\hat{\imath},\hat{\jmath}}}{A_{\hat{\jmath}}}, \text{ where } A_{\hat{\jmath}} = \sum_{i=1}^{p} a_{i,\hat{\jmath}} \quad . \tag{3}$$

A new oracle Θ (Fig.1(b)) is introduced. Given an observable O_S, it returns a (guessed) prediction $P_{m,k}$ with probability given by $\mathcal{P}(P_{m,k}|O_R)$, and with $O_R = \mathcal{R}(O_S, \hat{k})$. The probability (3) is then the probability of the oracle returning the correct prediction $P_{m,\hat{k}}$. This probability $\mathcal{P}$ is called the **matching probability** of O_S. The average number q' of queries to Θ required to gather q correct guesses is evaluated. The oracle Θ_m allows to evaluate the quality of an attack path $\mathcal{R}_m$ with the model m. A smaller q means a better adequacy between the attack path and the model. $\mathcal{P}$ represents the quality of the measures O_R with respect to the predictions $P_{m,\hat{k}}$. Finally this probability and the oracle are combined to globally evaluate the experimental attack with respect to the models thanks to q'.

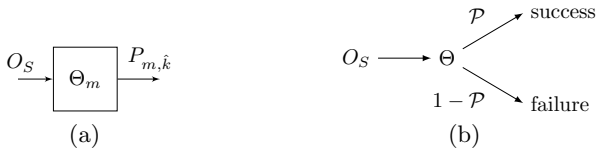

Fig. 1. (a): Oracle Θ_m. (b): Oracle with matching probability $\mathcal{P}$.

	$O_R = O_1$	$O_R = O_2$	$\cdots$	$O_R = O_o$	total
$P_{m,\hat{k}} = P_1$	$a_{1,1}$	$a_{1,2}$	$\cdots$	$a_{1,o}$	$\sum_{j=1}^{o} a_{1,j}$
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$
$P_{m,\hat{k}} = P_p$	$a_{p,1}$	$a_{p,2}$	$\cdots$	$a_{p,o}$	$\sum_{j=1}^{o} a_{p,j}$
total	$A_1 = \sum_{i=1}^{p} a_{i,1}$	$A_2 = \sum_{i=1}^{p} a_{i,2}$	$\cdots$	$A_o = \sum_{i=1}^{p} a_{i,o}$	n

Table 1. Contingency table of the measured O_R and the predicted values $P_{m,\hat{k}}$.

4 Practical Examples

Targeted algorithm: Advanced Encryption Standard (AES). The AES is a standard established by the NIST [23]. It is a block cipher. The encryption first consists in mapping the plaintext T of 128 bits into a two-dimensional array of bytes, called the State. Then, after a preliminary XOR between the input and the key, AES_{128} executes 10 times a round function that operates on the State. The operations used during these rounds are: **SubBytes** composed of non-linear transformations: 16 s-boxes noted SB; **ShiftRows** (SR), a byte shifting operation on each row of the State; **MixColumns** (MC), a linear matrix multiplication working on each column of the State; and a byte-wise xor $\oplus$ between the State and K_r, $r \in [\![0, 10]\!]$, the derived key used at round r.

Targeted device. In this paper, the target is the cipher key of an AES_{128}. It is implemented on an ARM-based STM32F100RB micro-controller embedding a Cortex-M3 core and running in our case at 24MHz. The board used is the STM32VLDICOVERY. This chip does not embed any countermeasures against physical attack but it is a popular choice for IoT applications.

4.1 Differential Power Analysis

Experimental protocol. An electromagnetic emissions analysis bench is composed of an EM probe from Langer (RF-R0,3-3) to capture the leakage, a preamplifier from Langer (PA 303) and an oscilloscope to measure it. The oscilloscope is a DSOS404A from Keysight. It achieves 10-bit resolution with a 20 Giga samples per second and 4GHz bandwidth. Finally, a control computer is used to orchestrate the measurements and perform the analysis.

Description of the attack. The first attack presented is the classic DPA/CPA attack [3] by electromagnetic analysis on the first round of the AES. The target is a byte $\hat{k}$ of the key K_0. There are only 256 possible values. The observable stimuli is a plaintext byte $O_S = T$. The reaction is the measured electromagnetic field, O_R = EM traces. The attack path is illustrated in Fig. 2 (left). The

theoretical path uses Hamming Weight (HW) as model.

$$\mathcal{R}(T, \hat{k}) = f(SB(T \oplus \hat{k}) \quad , \text{ with } f \text{ the physical function.}$$
$$\mathcal{R}_m(T, k) = HW(SB(T \oplus k)) = P_{m,k} \quad .$$

The distinguisher can be a difference of mean, a correlation, a mutual information, a principal component or a linear discriminant. **Evaluation of the attack.** The oracle Θ_m returns a Hamming weight. On average, $q = 4$ queries to Θ_m are required to retrieve a key byte. Then the oracle Θ is called. It returns a guessed Hamming weight. With the measures O_R collected in this experiment, an average of $q' = 17.8$ calls to Θ are required to have $q = 4$ correct guesses.

4.2 Differential Fault Analysis

Experimental protocol. The fault injection bench used electromagnetic pulses. These electromagnetic pulses are sent to the target thank to the inductive coupling of an EM probe with the target metal layers. Our bench is able to inject a pulse of ≈ 3 ns at the minimum and to repeat this pulse in order to achieve multi-faults if wanted.

Description of the attacks. The second attack presented is an attack of type DFA. Our bench can produce faults at the end of the round 9. We wanted to realize the attack of Giraud [8]. The target is a byte $\hat{k}$ of the key K_{10}. There are only 256 possible values. The stimuli is a ciphertext byte $O_S = C$. The reaction is a faulty ciphertext byte $O_R = C^{*}$[6]. The attack path is illustrated in Fig 2 (right). The theoretical attack path uses a single-bit fault model, *i.e* 8 possible models: $\oplus 2^i$ with $i \in [\![0, 7]\!]$, are considered together[7].

$$\mathcal{R}(C, \hat{k}) = SB\left(f\left(SB^{-1}\left(C \oplus \hat{k}\right)\right)\right) \oplus \hat{k} \quad \text{,with } f \text{ the fault injection process.}$$
$$\mathcal{R}_m(C, k) = SB\left(\left(SB^{-1}\left(C \oplus k\right)\right) \oplus 2^i\right) \oplus k \quad .$$

In the case of the Giraud's attack, the distinguisher could be a sieve [7] or a counter [8]. Unfortunately with our experimental protocol we cannot obtain single-bit faults , therefore the Non-Uniform Error Value Analysis (NUEVA) attack from [24] is chosen. The main idea in this attack comes from the fact that fault injection are never random. The stimuli is a pair of a ciphertext byte and a faulty ciphertext byte $O_S = (C, C^{*})$. The reaction is the error observed, $O_R = e$. The attack path ans theoretical attack path are:

$$\mathcal{R}\left((C, C^{*}), \hat{k}\right) = f\left(SB^{-1}\left(C \oplus \hat{k}\right), SB^{-1}\left(C^{*} \oplus \hat{k}\right)\right) \quad ,$$
$$\mathcal{R}_m\left((C, C^{*}), k\right) = SB^{-1}\left(C \oplus k\right) \oplus SB^{-1}\left(C^{*} \oplus k\right) \quad .$$

The model is $\oplus e$ (256 models). The distinguisher is an entropy.The goal is to detect if the distribution of the errors is uniform or not.

[6] In this paper a faulty variable is denoted by an asterisk *.
[7] The function SR shift bytes, so it is omitted for the simplicity of the equation.

Evaluation of the attacks In the case of Giraud attack, the oracle Θ_m is build with 8 models, so it returns 8 guess values. In average, $q = 2.4$ queries are necessary to retrieve one byte (result given in [8]). This model is very good but very hard to realize in practice with our fault injection bench. In the case of NUEVA, the oracle Θ_m returns an error. In theory, an infinity of queries ($q = \infty$) are necessary to evaluate if a distribution is uniform or not. Θ returns a guessed error. A faulty value can always be represented with a $\oplus$, so the matching probability is always equal to 1. So $q' = q = \infty$, and this model seems very bad. But in practice with our fault injection bench, 2500 faults are required in average.

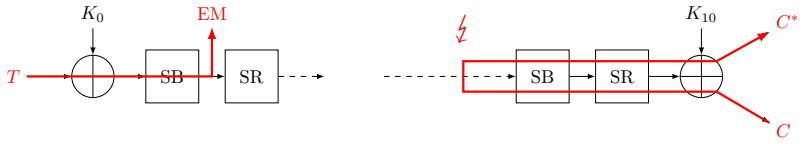

Fig. 2. Attack path of DPA (left) and DFA (right).

5 Conclusion

A new technique to evaluate physical attacks has been presented. The main idea is to evaluate the attack at different levels and not only on the final result. In a first time, only the models are studied, looking at the average number q of queries to an oracle that always returns the correct prediction. A smaller q means a better theoretical attack path. Then in a second time, only the predictions and the reactions are confronted, before using a distinguisher. Another oracle is introduced that returns a prediction but, contrary to the previous oracle, may return an incorrect one. The distribution of the returned predictions depends on the measures collected. The average number q' of queries to the new oracle to have q correct predictions evaluates the quality of the model with respect to the measures. At the end of this two-step evaluation, different distinguishers can be tested and a success rate can be computed. In case of failure, the advantage of our tool is to underline what part of an attack is weak.

References

1. Stefan Mangard. A simple power-analysis (SPA) attack on implementations of the AES key expansion. In *ICISC 2002*. Springer.
2. Paul C. Kocher and Joshua Jaffe and Benjamin Jun. Differential Power Analysis. In *CRYPTO*, 1999.
3. Eric Brier, Christophe Clavier and Francis Olivier. Correlation Power Analysis with a Leakage Model. In *CHES*, 2004.
4. Paul C Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *CRYPTO*. Springer, 1996.
5. Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power analysis attacks: Revealing the secrets of smart cards*, volume 31. Springer Science & Business Media, 2008.

6. Jean-Jacques Quisquater and David Samyde. Electromagnetic analysis (EMA): Measures and counter-measures for smart cards. In *Smart Card Programming and Security*. Springer, 2001.
7. Eli Biham and Adi Shamir. Differential Fault Analysis of Secret Key Cryptosystems. In *CRYPTO*, 1997.
8. Christophe Giraud. DFA on AES. In *Advanced Encryption Standard - AES*. Springer, 2005.
9. Sergei P Skorobogatov and Ross J Anderson. Optical fault induction attacks. In *Cryptographic Hardware and Embedded Systems-CHES 2002*. Springer, 2003.
10. Loïc Zussa, Jean-Max Dutertre, Jessy Clédière and Assia Tria. Power supply glitch induced faults on FPGA: An in-depth analysis of the injection mechanism. In *IOLTS*, 2013.
11. Nicolas Moro, Amine Dehbaoui, Karine Heydemann, Bruno Robisson, and Emmanuelle Encrenaz. Electromagnetic fault injection: towards a fault model on a 32-bit microcontroller. In *FDTC*. IEEE, 2013.
12. Silvio Micali and Leonid Reyzin. Physically Observable Cryptography. In *Theory of Cryptography Conference, TCC, Proceedings*. Springer, 2004.
13. François-Xavier Standaert, François Koeune and Werner Schindler . How to compare profiled side-channel attacks? In *ACNS*. Springer, 2009.
14. Abdelaziz M Elaabid and Sylvain Guilley. Practical improvements of profiled side-channel attacks on a hardware crypto-accelerator. In *AFRICACRYPT*. Springer, 2010.
15. Stefan Mangard, Elisabeth Oswald, and François-Xavier Standaert. One for all - all for one: unifying standard differential power analysis attacks. *IET Information Security*, 5(2), 2011.
16. Carolyn Whitnall, Elisabeth Oswald, and François-Xavier Standaert. The myth of generic dpa...and the magic of learning. In *CT-RSA, Proceedings*. Springer, 2014.
17. Houssem Maghrebi, Olivier Rioul, Sylvain Guilley and Jean-Luc Danger. Comparison between Side-Channel Analysis Distinguishers. In *ICICS*, 2012.
18. Ingrid Verbauwhede, Dusko Karaklajic and Jörn-Marc Schmidt. The Fault Attack Jungle-A Classification Model to Guide You. In *FDTC*, 2011.
19. Amir Moradi, Mohammad T Manzuri Shalmani and Mahmoud Salmasizadeh. A generalized method of differential fault attack against AES cryptosystem. In *Cryptographic Hardware and Embedded Systems-CHES 2006*. Springer.
20. Kazuo Sakiyama and Yang Li and Mitsugu Iwamoto and Kazuo Ohta. Information-Theoretic Approach to Optimal Differential Fault Analysis. *IEEE Transactions on Information Forensics and Security*, 7(1), 2012.
21. François-Xavier Standaert, Tal Malkin and Moti G Yung. A unified framework for the analysis of side-channel key recovery attacks. In *Advances in Cryptology-Eurocrypt 2009*. Springer.
22. Suresh Chari, Josyula R Rao, and Pankaj Rohatgi. Template attacks. In *Cryptographic Hardware and Embedded Systems-CHES 2002*. Springer, 2003.
23. NIST. Specification for the Advanced Encryption Standard. *FIPS PUB 197*, 2001.
24. Ronan Lashermes, Guillaume Reymond, Jean-Max Dutertre, Jacques Fournier, Bruno Robisson and Assia Tria. A DFA on AES Based on the Entropy of Error Distributions. In *FDTC*, 2012.